Reinforcement learning of recurrent neural network for temporal coding

Daichi Kimura Yoshinori Hayakawa*

Abstract

We study a reinforcement learning for temporal coding with neural network consisting of stochastic spiking neurons. In neural networks, information can be coded by characteristics of the timing of each neuronal firing, including the order of firing or the relative phase differences of firing. We derive the learning rule for this network and show that the network consisting of Hodgkin-Huxley neurons with the dynamical synaptic kinetics can learn the appropriate timing of each neuronal firing. We also investigate the system size dependence of learning efficiency.

1 Introduction

Many studies have assumed that neurons transmit information by their firing rate. The McCulloch-Pits unit is a typical model and networks of these units have been investigated. On the other hand, recent experiments suggest that the timing of neuronal firing may also contribute to the information representation function in the brain and the synaptic modification (Gray, König, Engel, & Singer, 1989; Bi & Poo, 1999; Varela, Lachaux, Rodriguez, & Martinerie, 2001; Reyes, 2003). For example, it seems that local and global synchronization play a significant role in integration of information which is distributed across the brain. Another example shows that the order of timing of neuronal firings can encode the information of stimuli on fingertips, and this encoding by sequence can transmit information faster than coding from the firing rate directly (Johansson & Birznieks, 2004).

To capture the dynamical aspects of neural networks, networks consisting of various model neurons other than the McCulloch-Pitts unit have been

*Department of Physics, Tohoku University, Sendai 980-8578, Japan

investigated, because McCulloch-Pitts units cannot describe the temporal behavior of neurons over short time scales. In this context, an associative memory for neural networks of oscillator neurons or spiking neurons has been studied (Aoyagi, 1995; Yoshioka & Shiino, 1998; Hoppensteadt & Izhikevich, 1999; Kanamaru & Okabe, 2000; Hasegawa, 2001; Lee & Farhat, 2001, 2002). In these systems, the relative phase differences, i.e., the timing of firings, are used to represent the memory.

There are few studies of learning in pulse neuron models such as those consisting of Hodgkin-Huxley (HH) neurons because of difficulty in deriving the learning rule. Although several studies have been made of learning in networks that consist of integrate-and-fire (IF) model neurons (Seung, 2003; Xie & Seung, 2004; Cios, Swiercz, & Jackson, 2004), most of these studies focus only on coding in terms of the firing rate.

However, it would be useful to combine temporal coding and learning because it has been shown that temporal coding can deal with more information and process it faster than coding from just the firing rate (Thorpe, Delorme, & Rullen, 2001). As an example of this, Delorme et al.(2001) show that a neural network consisting of IF neurons can learn to identify human faces by using “Rank Order Coding”, i.e., coding by the order of timing of each neuronal firing, where neurons are allowed to spike once only.

In this paper, we study a reinforcement learning for temporal coding with neural network consisting of stochastic spiking neurons. After defining a network of coupled stochastic HH neurons and some quantities in Sec. II, we train the network to learn an XOR operation, where the output information is coded by the order of firing in Sec. III. In Sec. IV, we investigate how the result or performance of learning depends on the system size and the strength of noise, and conclusions follow.

2 The model

To illustrate an example of the learning process of spiking neurons, we consider a neural network consisting of HH neurons. Since HH neurons show excitability, they can code information by the timing of firing. The complete dynamics for a network of coupled HH neurons may be expressed as

$$C_m \frac{dV_i}{dt} = g_{Na} m_i^3 h_i (V_{Na} - V_i) + g_K n_i^4 (V_K - V_i) + g_L (V_L - V_i) \\ + \sum_j w_{ij} I_j^s(t) + \xi_i(t), \tag{1}$$

$$\frac{dm_i}{dt} = \alpha_m(V_i)(1 - m_i) - \beta_m(V_i) m_i, \tag{2}$$

$$\frac{dh_i}{dt} = \alpha_h(V_i)(1 - h_i) - \beta_h(V_i) h_i, \tag{3}$$

$$\frac{dn_i}{dt} = \alpha_n(V_i)(1 - n_i) - \beta_n(V_i) n_i, \tag{4}$$

where V_i is the membrane potential of neuron i, C_m the membrane capacitance, $V_r (r = \text{Na,K,L})$ are the equilibrium potentials, $g_r (r = \text{Na,K,L})$ the conductance, m_i, h_i, n_i the voltage dependent activating or inactivating variables, α_x and $\beta_x (x = m, h, n)$ the functions of voltage V_i (Hodgkin & Huxley, 1952), w_{ij} is the synaptic weight from neuron j to i ($w_{ij} \neq w_{ji}$ in general), $I_j^s(t)$ the synaptic current and $\xi_i(t)$ is the Gaussian white noise which obeys,

$$\overline{\xi_i(t)} = 0, \tag{5}$$

$$\overline{\xi_i(t)\xi_j(t')} = Q \delta_{ij} \delta(t - t'), \tag{6}$$

where $\overline{A}$ is the average of A over time and Q the variance of noise. The synaptic current $I_j^s(t)$ is given by

$$I_j^s(t) = r_j(t)[V_{syn} - V_i], \tag{7}$$

where V_{syn} is the synaptic reversal potential and $r_j(t)$ the fraction of bound receptors (Destexhe, Mainen, & Seijnowski, 1994) described by

$$\frac{dr_j}{dt} = \alpha T(t)(1 - r_j) - \beta r_j, \tag{8}$$

$$T(t) = \begin{cases} 1 & t_j^0 \leq t < t_j^0 + \tau, \\ 0 & \text{otherwise}, \end{cases} \tag{9}$$

where $\alpha = 0.94$ ms^{-1}, $\beta = 0.18$ ms^{-1}, t_j^0 the time when the presynaptic neuron j fires (membrane potential over 27 mV) and $\tau = 1.5$ ms (Fernández, Huerta, Corbacho, & Sigüenza, 2000) Fig. 1 shows the behavior of $V_i(t)$ and $r_i(t)$ of single neuron added the external current whose amplitude is 10 mA. We used the forth order Runge-Kutta method with the time step $\Delta t = 0.01$ to solve Eqs. (1)$\sim$(4).

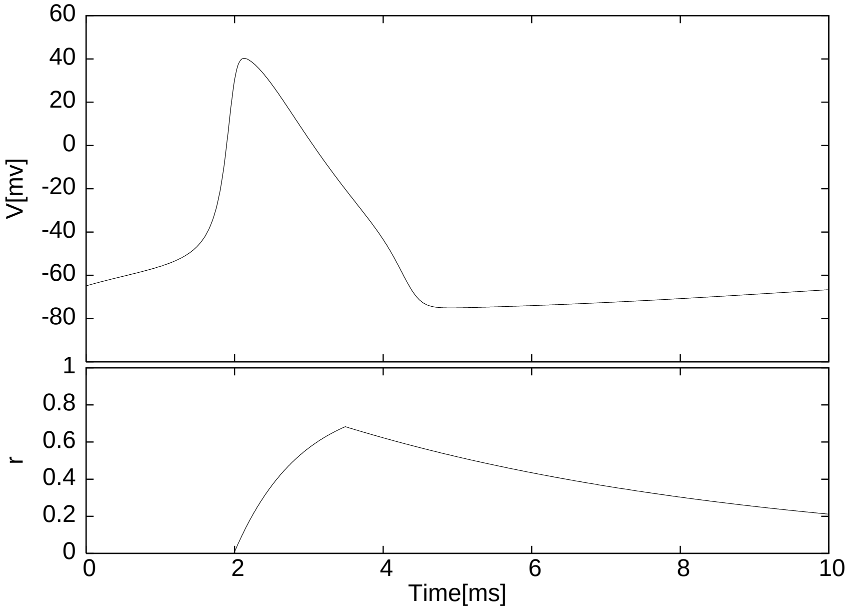

Figure 1: The behavior of $V_i(t)$ and $r_j(t)$ of single HH neuron added the external current. The neuron fires at $t = 2$ ms, then the value of $r_i(t)$ starts to increase. After a lapse of $\tau = 1.5$ ms, $r_i(t)$ turns into decline.

To train the neural network, we use a reinforcement learning algorithm. Let us consider time sequences of states of neurons; $\sigma \equiv (\boldsymbol{V}(0), \boldsymbol{V}(1), \boldsymbol{V}(2), \cdots, \boldsymbol{V}(T))$, where $\boldsymbol{V}(t)$ denotes the vector $(V_1(t), \cdots, V_N(t))$. We assign a scalar value ("reward") to each time sequence σ according to the signal from the network (Sutton & Barto, 1998). We give a high reward R to the desirable time sequence σ in each episode. Here we consider episodic learning. Since Eq. (1) includes the Gaussian white noise, we calculate the expected value of the reward $\langle R \rangle$, where $\langle \cdots \rangle$ signifies the average over all possible time sequences σ. Then the goal of learning is to maximize $\langle R \rangle$ by adjusting w_{ij}. We use an ascending gradient strategy;

$$w_{ij}^{\mathrm{New}} = w_{ij}^{\mathrm{Old}} + \delta w_{ij}, \tag{10}$$

$$\delta w_{ij} = \epsilon \frac{\partial \langle R \rangle}{\partial w_{ij}}, \tag{11}$$

where ϵ is the learning coefficient. We can calculate the gradient of $\langle R \rangle$ with respect to w_{ij} (Hayakawa, 2001; Fiete & Seung, 2006),

$$\frac{\partial \langle R \rangle}{\partial w_{ij}} = \frac{1}{Q} \left\langle R(\sigma) \int_0^T dt \xi_i(t) I_j^s(t) \right\rangle. \tag{12}$$

For details of the derivation, see appendix.

3 Learning procedure for temporal coding

The present learning rule Eq. (10)$\sim$(12) can be effective for any information coding including the order coding. To show an example, we consider a neural network consisting of 2 input neurons, 15 output neurons and hidden neurons. We divide the set of output neurons into three disjoint subsets, $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$, each containing 5 output neurons.

Here, we assume that information is coded by the temporal order of the "group activity" of subsets $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$. As a learning goal, we chose an XOR operation in terms of order coding; the subsets $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ should fire in this order for input patterns $[-1, -1]$ and $[1, 1]$, and in reverse order for input patterns $[-1, 1]$ and $[1, -1]$. We train the neural network to learn the suitable timing t_p^k of firing in the subset $\mathcal{O}_p$ for given input pattern $k = [-1, -1], [-1, 1], [1, -1]$ and $[1, 1]$, respectively, as described in Table 1. The right column of Table 1 shows the desirable order of group activity of subsets for each input pattern. The learning goal is the reduction of the RMS

Table 1: The order of collective firing as a learning goal. $t_1 = 2.0$, $t_2 = 4.0$ and $t_3 = 6.0$, respectively.

Input	t_p^k			Desirable
pattern	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	order
$[-1, -1]$	t_1	t_2	t_3	$\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$
$[1, -1]$	t_3	t_2	t_1	$\mathcal{O}_3$, $\mathcal{O}_2$, $\mathcal{O}_1$
$[-1, 1]$	t_3	t_2	t_1	$\mathcal{O}_3$, $\mathcal{O}_2$, $\mathcal{O}_1$
$[1, 1]$	t_1	t_2	t_3	$\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$

errors E^k for each input pattern k given by

$$E^k = \frac{1}{N_{\mathrm{O}}} \sum_{p=1}^{3} \sum_{i \in O_p} \left(t_i^0 - t_p^k \right)^2 , \tag{13}$$

where t_i^0 is the time when the neuron i fires and N_{O} the number of output neurons.

We define an "episode" as the transient dynamics of the network which lasts from $t = 0$ to $t = T$ accompanied with external inputs. We evaluate the reward R depending on the resulting error E^k through each episode for given input pattern as

$$R = \begin{cases} 1 & E^k < E_{\mathrm{th}}^k, \\ 0 & E^k \geq E_{\mathrm{th}}^k, \end{cases} \tag{14}$$

where E_{th}^k is a threshold of error. We change the threshold E_{th}^k gradually as the learning proceeds. In the following simulations, we calculate the average error E_{ave}^k over 100 episodes for each input pattern, and we update the threshold for the next 100 episodes by $E_{\mathrm{th}}^k = 0.99 \times E_{\mathrm{ave}}^k$. E_{th}^k and E_{ave}^k are almost identical. It sometimes happens that E_{ave}^k increases through learning since the update of w_{ij} to decrease E^k will affect the value of $E^{k'}$ where $k \neq k'$. Thus, E_{th}^k also sometimes increases, not decreases monotonically.

During each episode, the input signals are fixed to the time-independent value, $I_i^s = I_0$ or $-I_0$, corresponding to the input patterns. (Here, $I_0 = 10$ mA.) The learning coefficient, the variance of noise and the period of one episode are $\epsilon = 0.001$, $Q = 10000$ and $T = 10.0$ ms, respectively.

We assume that all the output and hidden neurons are excitatory, i.e., w_{ij} for $j \notin \mathcal{I}$ can take only positive value, where $\mathcal{I}$ is the set of input neurons, while w_{ij} for $j \in \mathcal{I}$ can take both positive and negative value. We first set the synaptic weights w_{ij} for $j \in \mathcal{I}$ using the uniformly distributed random number in $[-1.0, 1.0]$, while for $j \notin \mathcal{I}$ between $[\frac{2.5}{N}, \frac{5.0}{N}]$, where N is the number of neurons. Due to this normalization factor $\frac{1}{N}$, the magnitude of error E^k does not depend on N. In this section, the number of hidden neurons was 5.

Fig. 2 shows the behavior of the output neurons at initial state (upper) and that after 5×10^4 episodes for each input pattern (that is, 2×10^5 episodes in total) elapsed (lower). We find that neurons in $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ fire in this order for the input patterns $[-1, -1]$ and $[1, 1]$, while in reverse order for the patterns $[1, -1]$ and $[-1, 1]$ after learning. In fact, Fig. 3 shows that the average error E^k_{ave} decrease gradually through learning. These figures indicate that learning in terms of the order of firing by applying the present learning rule is successful.

Fig. 4 shows w_{ij} in a matrix form representing the connections with color after learning. As can be seen, there is few connection between $\mathcal{O}_1$ and $\mathcal{O}_3$ as a result of learning. This can be interpreted as the consequence that there is no direct correlation between $\mathcal{O}_1$ and $\mathcal{O}_3$ in the both of the output time sequence. Few connections from $\mathcal{I}$ to $\mathcal{O}_2$ can be understood in the same way.

4 System size dependence of learning efficiency

Since realistic biological systems consist of many neurons, it is interesting and important how efficient this learning rule is for larger system size. In this section, we discuss the system size dependence of the performance of learning

At first, we train the network for various number of hidden neurons N_{H}. In the following simulation, all parameters but N_{H} are given as the same as previous section. The sum of average errors $\sum_k E^k_{\text{ave}}$ are shown in Fig. 5(a). To investigate the property of learning, we assumed an exponential decay of the error $\sim \exp(-\lambda s)$ as the function of the number of episode s and showed λ in inset of Fig. 5(a). As shown in Fig. 5(a), the speed of learning becomes slower when extra hidden neurons are added, and, decay rate λ also has the same tendency. However, the error is significantly reduced as the episodes proceed. In this case, it turned out that the network with $N_{\text{H}} = 5$ showed best result in this simulation so that those added neurons are redundant and do not contribute learnability.

Next, we examine the case the learning speed for various number of output neurons N_{O}, where each subsets $\mathcal{O}_p$ consist of $N_{\text{O}}/3$ and the number of hidden neurons $N_{\text{H}} = 5$. Fig. 5(b) shows the error $\sum_k E^k_{\text{ave}}$ and the decay rate λ (inset). In this first stage of the learning process ($s \leq 10^4$), decay rate is approximately given as $\lambda \sim N_{\text{O}}^{\delta}$, where $\delta \approx -2$. However, if λ is evaluated

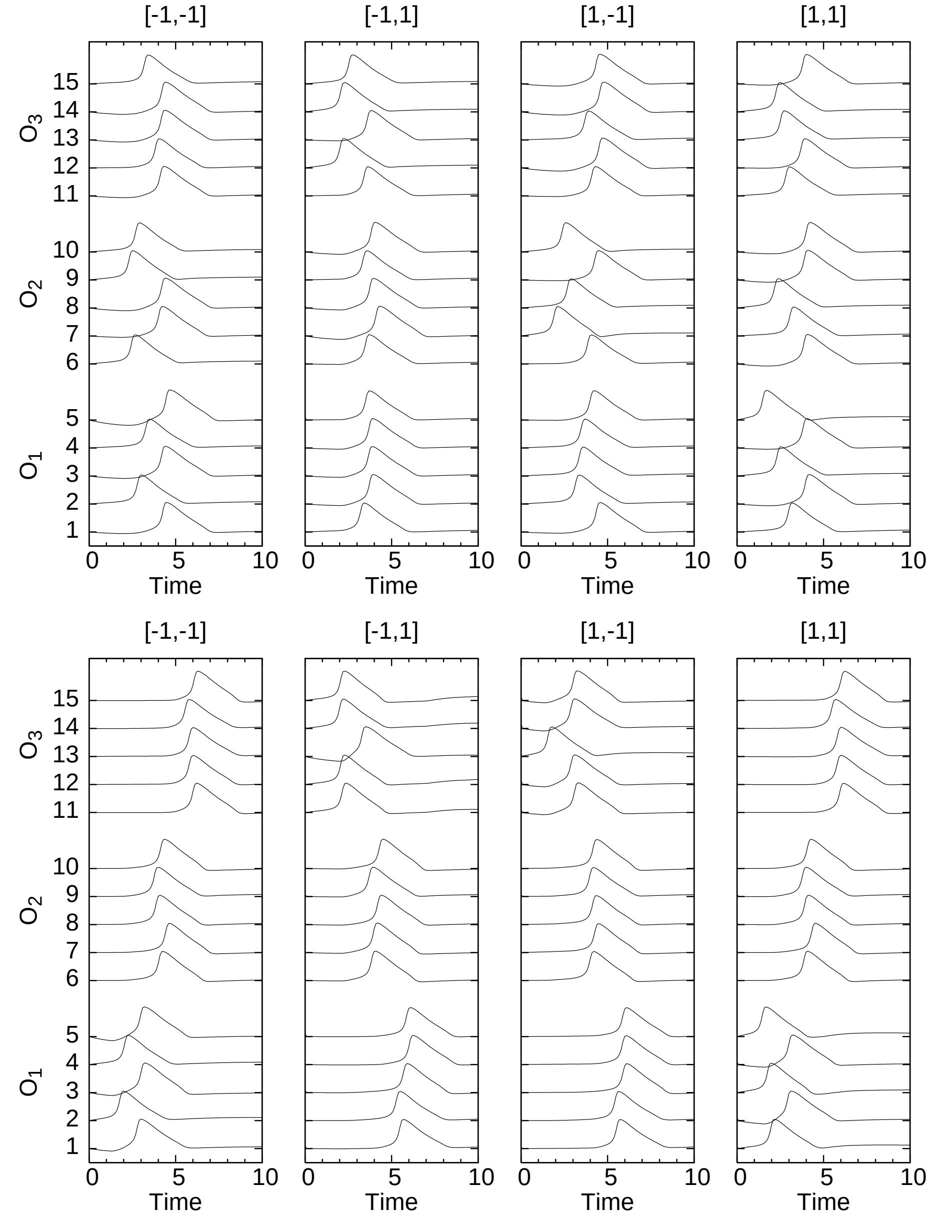

Figure 2: The action potential of each output neuron at initial state (upper) and after 5×10^4 episodes for each input pattern (that is, 2×10^5 episodes in total) elapsed (lower). The index of each output neuron is shown on the vertical axis. The neurons with index ≤ 5 are in $\mathcal{O}_1$, those with index ≥ 11 are in $\mathcal{O}_3$, and the others are in $\mathcal{O}_2$. $\mathcal{O}_1$, $\mathcal{O}_2$ and $\mathcal{O}_3$ fire in this order for input patterns [-1,-1] and [1,1], while in reverse order for [1,-1] and [-1,1]. See also Table 1.

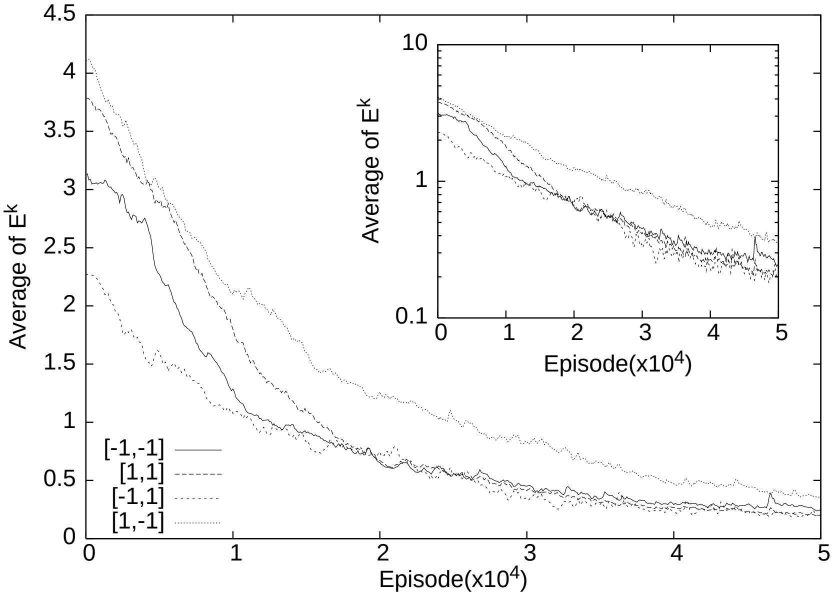

Figure 3: E^k_{ave} for each input pattern as a function of episodes through learning. Inset is the same data in semi-logarithmic plot. Typical value of E^k before learning is approximately 3.

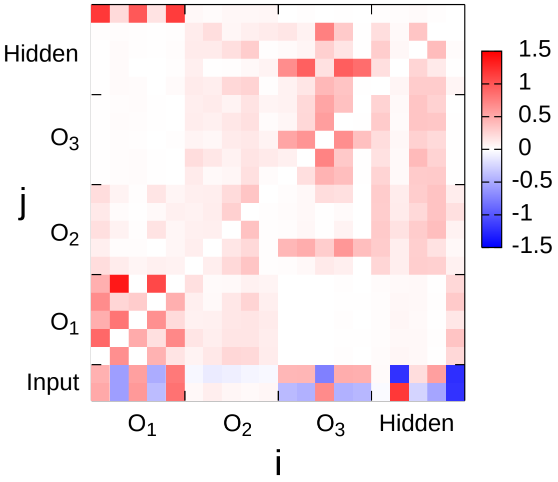

Figure 4: w_{ij} in a matrix form representing the connections with color after learning. Neuron j is presynaptic and i is postsynaptic.

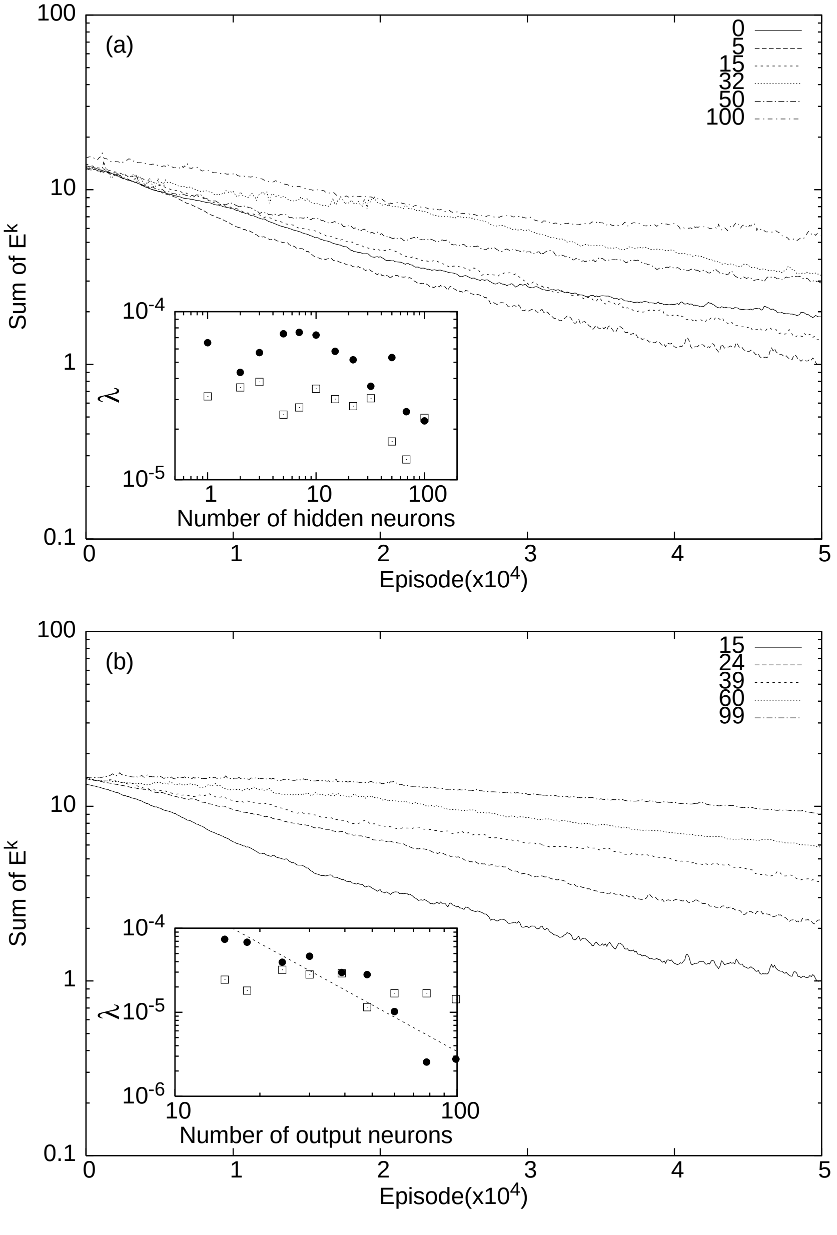

Figure 5: $\sum_k E_{\rm ave}^k$ as a function of episodes through learning for each $N_{\rm H}$(a) and $N_{\rm O}$(b). Inset: The decay constant λ of $\sum_k E_{\rm ave}^k$ for $0 < s < 10^4$ (filled circle) and $4 \times 10^4 < s < 5 \times 10^4$ (open square) where s is the number of episode. The slope of dotted line in inset (b) is approximately -2.

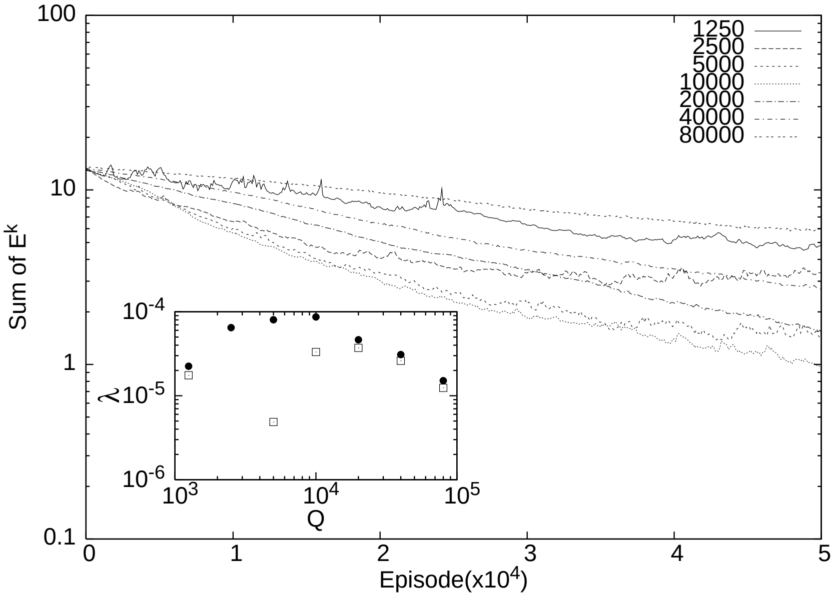

Figure 6: $\sum_k E^k_{\text{ave}}$ as a function of episodes through learning for each Q. Inset: λ of $\sum_k E^k_{\text{ave}}$ for $0 < s < 10^4$ (filled circle) and $4 \times 10^4 < s < 5 \times 10^4$ (open square).

in the interval $4 \times 10^4 < s < 5 \times 10^4$, the N_{O} dependence of λ seems to be more weaken, so that a long-term behavior of the learning process needs to be investigated to discuss the asymptotics.

We would like to mention the effects of noise in the learning. The noise strength may affect the result or performance of learning because a random search process in the weight space is the essential part of this learning algorithm. We tested the dependence of the learning performance on the variance of noise Q in Eq. (6) , where $N_{\text{O}} = 15$, $N_{\text{H}} = 5$ and other parameters are given as previous section. Fig. 6 shows the learning error and its decay rate. Since the amount of update $|\delta w_{ij}|$ is in the order of $|\epsilon/Q|$, the magnitude of Q will directly affect the stability of learning. However, Fig. 6 indicates this learning rule is robust against the change of Q as long as the gradient ascending process assumed in Eq. (11) holds.

5 Conclusions

We have proposed a concrete method to encode temporal information for a network of stochastic spiking neurons utilizing a reinforcement learning method, and we show that this learning rule is effective for order coding.

Since the derivation of Eq. (11) does not depend on the form of the differential equation for the action potential like Eq. (1), the learning rule Eq. (11) is generally applicable to networks of other types of model neurons, for example neurons following the IF or FitzHugh-Nagumo (FHN) model with random fluctuation. In fact, we have checked the network consisting of IF or FHN neurons can also learn the same order coding task with the learning rule Eq. (11). (We omit these details in this paper.)

The present learning rule can be applied not only to the order coding, but also to any information coding. In general, if one can design an appropriate reward R by defining the error of group activity from the reference signals, the learning process is given in a straightforward manner as described in Sec. III. For example, we confirmed that the present method also worked for the XOR task in terms of phase coding, where the relative phase difference codes information in the same way as oscillator neurons (Aoyagi, 1995). In the simulation, neural network consisted of 2 input neurons, a few hidden neurons and 10 output neurons which were divided into two disjoint subsets, $\mathcal{O}_1$ and $\mathcal{O}_2$, each containing 5 output elements. Learning goal was for group activity in $\mathcal{O}_1$ and $\mathcal{O}_2$ to become in phase for input patterns $[-1,-1]$ and $[1,1]$, and to be out of phase for $[1,-1]$ and $[-1,1]$.

In general, convergence of the reinforcement learning is slow and the present model is not the exception. Several factors need to be investigated to improve the speed of convergence. Although we have assumed Gaussian white noise in this study, in real neuronal systems it is likely that noise may obey other statistical properties. For example, Fiete et al. (2006) discusses a learning rule using the arbitrary noise. Furthermore, learning speed depends to a great extent on the design of the cost function, i.e., the reward. If the design of the reward is not suitable, the learning process drops into local minima easily. In the present study, we empirically employed a sliding threshold method by changing the learning goal E^k_{th} in Eq. (14) started from easier one. Although this strategy can be applied to any learning tasks, the performance or result of learning is sensitive to define E^k_{th}. In this paper, we have defined E^k_{th} as $A \times E^k_{\mathrm{ave}}$ where $A = 0.99$, because $A = 0.9$ is too severe to learn so that the speed of learning becomes slower and $A = 1.0$ causes overestimated reward so that the learning process drops into local minima. The point that E^k_{th} does not monotonically decrease is also important since the update of w_{ij} to decrease E^k will affect the value of $E^{k'}$ where $k \neq k'$. Therefore, if we decrease E^k_{th} monotonically while E^k_{ave} increases because of this influence with update of w_{ij}, the task becomes harder to learn. As well as the conventional back propagation learning, how to decide the number of hidden neurons and assign the initial value of w_{ij} are also important problem.

In the XOR task with the order coding, convergence speed in learning process significantly affected by both the number of hidden and output neurons, larger size system seems to require more learning steps. We intend to examine this relationship in future research.

We have assumed a fully connected network and assigned the random value to the initial state of w_{ij}, which may not be appropriate for learning. Watts and Strogatz have proposed a "small-world" network structure (Watts & Strogatz, 1998). This network has been investigated in the context of a multi-layered feed-forward network, and some improvements in performance have been made (Simard, Nadeau, & Kröger, 2005). Also, the propagation velocity and coherent oscillation of IF or Hodgkin-Huxley neurons depend on the topology of network (which includes small-world network) (Fernández et al., 2000; Roxin, Riecke, & Solla, 2004). These results may indicate that there are some suitable topologies of network for temporal coding as well. Since the present learning rule Eq. (12) includes only the local relation between neuron i and j, this learning rule may be applied to a network with an arbitrary topology. Finding a suitable network topology to facilitate learning is also a problem for future investigation.

Acknowledgments

We would like to thank Dr. Toshihiro Kawakatsu and Dr. Tsuyoshi Hondou for helpful comments.

Appendix : Estimation of gradient of $\langle R \rangle$

Let us introduce a stochastic processing element as a model neuron. The value of action potential of i-th neuron at discrete time $t(= 0, 1, 2, \cdots)$ is represented by $V_i(t)$. $V_i(t)$ and other internal state $m_i(t)$ at successive time step $t+1$ are then determined as

$$V_i(t+1) = g(V_i(t), m_i(t)) + \sum_j^N w_{ij} f(V_j(t)) + \xi_i(t), \qquad (15)$$

$$m_i(t+1) = h(V_i(t), m_i(t)), \qquad (16)$$

where w_{ij} is the synaptic weight from neuron j to i, $f(V_i(t))$ the signal which i-th neuron emits according to $V_i(t)$, N the total number of neurons, $g(V_i(t), m_i(t))$ and $h(V_i(t), m_i(t))$ are functions of the neuronal states, and $\xi_i(t)$ is the Gaussian white noise which obeys Eq. (5) and Eq. (6). We consider that $f(V_i(t))$, $g(V_i(t), m_i(t))$ and $h(V_i(t), m_i(t))$ are arbitrary functions.

As a continuous time representation, we employ a Langevin equation for neuronal updates;

$$\frac{dV_i}{dt} = g(V_i(t), m_i(t)) + \sum_j^N w_{ij} f(V_j(t)) + \xi_i(t), \tag{17}$$

$$\frac{dm_i}{dt} = h(V_i(t), m_i(t)). \tag{18}$$

Although we treat $m_i(t)$ as a scalar variable in the following discussion, the same method can also be applied if $m_i(t)$ is multivariate.

We consider time sequences of states of neurons for $t = 0, \cdots, T$; $\sigma \equiv (\boldsymbol{V}(0), \boldsymbol{V}((1), \boldsymbol{V}(2), \cdots, \boldsymbol{V}(T))$, where $\boldsymbol{V}(t)$ denotes the vector $(V_1(t), \cdots, V_N(t), m_1(t), \cdots, m_N(t))$.

We assign a reward R to each time sequence σ and calculate the gradient of $\langle R \rangle$ with respect to w_{ij} (Hayakawa, 2001; Fiete & Seung, 2006). The probability density $\mathcal{T}_\sigma$ for each time sequence is defined as

$$\mathcal{T}_\sigma = P(\boldsymbol{V}(T)|\boldsymbol{V}(T-1)) \cdots P(\boldsymbol{V}(1)|\boldsymbol{V}(0)), \tag{19}$$

where $P(A|B)$ is the transition probability from state B to A. Since $m_i(t)$ is deterministic and only $V_i(t)$ includes the Gaussian noise described by Eqs. (5) and (6), the transition probability in Eq. (19) may be expressed as

$$\begin{aligned} P(\boldsymbol{V}(t+1)|\boldsymbol{V}(t)) &= \prod_i \frac{1}{(2\pi Q)^{1/2}} \exp\left[-\frac{1}{2Q}\xi_i^2(t)\right] \\ &\quad \times \delta_{m_i(t+1), h(V_i(t), m_i(t))} \\ &= \frac{1}{(2\pi Q)^{N/2}} \exp\left[-\frac{1}{2Q}\sum_i \eta_i(t)\right] \prod_i \delta_{m_i(t+1), h(V_i(t), m_i(t))}, \end{aligned} \tag{20}$$

where

$$\eta_i(t) \equiv \left[V_i(t+1) - g(V_i(t), m_i(t)) - \sum_j^N w_{ij} f(m_j(t))\right]^2. \tag{21}$$

and $\delta_{m,n}$ is the Kronecker delta. The average of a quantity $Y(\sigma)$ over all possible time sequences can be described by

$$\begin{aligned} \langle Y \rangle &= \frac{1}{Z}\sum_\sigma Y(\sigma)\mathcal{T}_\sigma \\ &= \frac{1}{Z}\sum_\sigma Y(\sigma) \exp\left[-\frac{1}{2Q}\sum_t \sum_i \eta_i(t)\right], \end{aligned} \tag{22}$$

where Z is a normalization factor. Therefore, the gradient of $Y(\sigma)$ may be obtained as

$$\begin{aligned}\frac{\partial\langle Y\rangle}{\partial w_{ij}} &= \frac{1}{Z}\sum_{\sigma}\mathcal{T}_{\sigma}Y(\sigma)\frac{1}{Q}\sum_{t=0}^{T}\xi_i(t)f(V_j(t)) \\ &= \frac{1}{Q}\left\langle Y(\sigma)\sum_{t=0}^{T}\xi_i(t)f(V_j(t))\right\rangle.\end{aligned} \tag{23}$$

In the continuous case (17), the sum over σ in Eq. (22) becomes the path integral

$$\langle Y\rangle = \frac{1}{Z}\int_{\sigma}Y(\sigma)\exp\left[-\frac{1}{2Q}\int_0^T dt\sum_i \tilde{\eta}_i(t)\right]d(\text{path}), \tag{24}$$

where

$$\tilde{\eta}_i(t) \equiv \left[\frac{dV_i}{dt} - g(V_i(t), m_i(t)) - \sum_j^N w_{ij}f(V_j(t))\right]^2. \tag{25}$$

Using $R(\sigma)$ as $Y(\sigma)$, we obtain the derivative in Eq. (11) as

$$\frac{\partial\langle R\rangle}{\partial w_{ij}} = \frac{1}{Q}\left\langle R(\sigma)\int_0^T dt\xi_i(t)f(V_j(t))\right\rangle. \tag{26}$$

For the network consisting of HH neurons (1)∼(4), we can derive Eq. (12) by instituting Eq. (7) to Eq. (26).

Thus, this result indicates that the calculation of the gradient of $\langle R\rangle$ depends only on $\xi_i(t)$, $f(V_j(t))$ and $R(\sigma)$. The linear summation of the presynaptic neuronal activity and an additive independent noise as in Eq. (15) and Eq. (17) is essential for the present learning algorithm. This learning rule includes only the local relation between neurons i and j, while backpropagation (Rumelhart, Hinton, & Williams, 1986) or RTRL (Williams & Zipser, 1989) requires the information of other neurons in addition to i, j in order to calculate δw_{ij}.

References

Aoyagi, T. (1995). Network of neural oscillators for retrieving phase information. *Phys. Rev. Lett.*, *74*, 4075.

Bi, G.-q., & Poo, M.-m. (1999). Distributed synaptic modification in neural networks induced by patterned stimulation. *Nature*, *401*, 792.

Cios, K., Swiercz, W., & Jackson, W. (2004). Networks of spiking neurons in modeling and problem solving. *Neurocomputing*, *61*, 99.

Delorme, A., & Thorpe, S. (2001). Face identification using one spike per neuron: resistance to image degradations. *Neural Networks*, *14*, 795.

Destexhe, A., Mainen, Z., & Seijnowski, T. (1994). An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, *6*, 14.

Fernández, L., Huerta, R., Corbacho, F., & Sigüenza, A. (2000). Fast response and temporal coherent oscillations in small-world networks. *Phys. Rev. Lett.*, *84*, 2758.

Fiete, I., & Seung, H. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Phys. Rev. Lett.*, *97*, 048104.

Gray, C., König, P., Engel, A., & Singer, W. (1989). Oscillatory responses in cat visual-cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, *338*, 334.

Hasegawa, H. (2001). An associative memory of hodgkin huxley neuron networks with willshaw-type synaptic couplings. *J. Phys. Soc. Jpn.*, *70*, 2210.

Hayakawa, Y. (2001). Reinforcement learning for stochastic neural networks. *Proceedings of 2001 International Symposium on Nonlinear Theory and its Applications*, *2*, 493.

Hodgkin, A., & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol.(London)*, *117*, 500.

Hoppensteadt, F., & Izhikevich, E. (1999). Oscillatory neurocomputers with dynamic connectivity. *Phys. Rev. Lett.*, *82*, 2983.

Johansson, R., & Birznieks, I. (2004). First spikes in ensembles of human tactile afferents code complex spatial fingertip events. *Nat.Neurosci.*, *7*, 179.

Kanamaru, T., & Okabe, Y. (2000). Associative memory retrieval induced by fluctuations in a pulsed neural network. *Phys. Rev. E*, *62*, 2629.

Lee, G., & Farhat, N. (2001). The bifurcating neuron network i. *Neural Networks*, *14*, 115.

Lee, G., & Farhat, N. (2002). The bifurcating neuron network 2: an analog associative memory. *Neural Networks*, *15*, 69.

Reyes, A. (2003). Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro. *Nat.Neurosci.*, *6*, 593.

Roxin, A., Riecke, H., & Solla, S. (2004). Self-sustained activity in a small-world network of excitable neurons. *Phys. Rev. Lett.*, *92*, 198101.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533.

Seung, H. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, *40*, 1063.

Simard, D., Nadeau, L., & Kröger, H. (2005). Fastest learning in small-world neural networks. *Physics Letters A*, *336*, 8.

Sutton, R., & Barto, A. (1998). *Reinforcement learning : An introduction.* Cambridge, MA: MIT Press.

Thorpe, S., Delorme, A., & Rullen, R. (2001). Spike-based strategies for rapid processing. *Neural Networks*, *14*, 715.

Varela, F., Lachaux, J., Rodriguez, E., & Martinerie, J. (2001). The brainweb: Phase synchronization and large-scale integration. *Nat.Rev.Neurosci.*, *2*, 229.

Watts, D., & Strogatz, S. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*, 440.

Williams, R., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*, 270.

Xie, X., & Seung, H. (2004). Learning in neural networks by reinforcement of irregular spiking. *Phys. Rev. E*, *69*, 041909.

Yoshioka, M., & Shiino, M. (1998). Associative memory based on synchronized firing of spiking neurons with time-delayed interactions. *Phys. Rev. E*, *58*, 3628.